

برنامه نویسی به زبان

# پایتون

مرجع کامل

تالیف

مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا  
مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از  
کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع  
کردن مولف از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرید

hadikiamarsi@gmail.com

لازم به ذکر است کلیه حقوق مادی و معنوی این اثر برای مولف محفوظ می  
باشد و هرگونه کپی برداری و استفاده از محتویات این کتاب به هر نوعی تحت  
پیگرد قانونی قرار می گیرد

# فصل دهم

**در این فصل مطالب زیر را خواهید آموخت**  
**دیکشنری ها ( Dictionaries ) در پایتون ( python )**  
**دسترسی به عناصر در دیکشنری ها ( Dictionaries )**  
**بروزرسانی دیکشنری ها ( Dictionaries )**  
**پاک کردن عناصر دیکشنری ها ( Dictionaries )**  
**خصوصیات کلیدهای دیکشنری ها ( Dictionaries )**  
**توابع و متدهای پیش ساخته برای دیکشنری ها ( Dictionaries )**

کیا مدتی

# دیکشنری ها ( Dictionaries ) در پایتون ( python )

یکی دیگر از انواع داده ای که در زبان برنامه نویسی پایتون ( python ) کاربرد فراوانی دارد دیکشنری ها ( dictionaries ) می باشند که در این فصل به تفصیل با آن ها آشنا می گردید . در دیکشنری ها ( dictionaries ) بجای استفاده از اندیس از کلید ها استفاده می گردد در دیکشنری ها ( dictionaries ) جفت کلید و داده داریم . جفت کلید و داده در دیکشنری ها ( dictionaries ) با علامت کولن : ( colon ) از هم جدا می گردند نشانه بارز دیکشنری ها ( dictionaries ) علامت { } می باشد

هر یک از داده های دیکشنری ها ( dictionaries ) می تواند رشته ، عدد و یا حتی تاپل باشد

## دسترسی به عناصر در دیکشنری ها ( Dictionaries )

برای دسترسی به عناصر یک دیکشنری ( dictionary ) نیاز به براکت باز و بسته به همراه نام کلید می باشد

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print "dict['Name']: ", dict['Name']
print "dict['Age']: ", dict['Age']
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
dict['Name']: Zara
dict['Age']: 7
```

اگر برای دسترسی به عناصر یک دیکشنری ( dictionary ) از کلیدی استفاده نمایید که وجود ندارد یک پیغام خطا ظاهر می گردد

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print "dict['Alice']: ", dict['Alice']
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
dict['Alice']:  
Traceback (most recent call last):  
  File "test.py", line 4, in <module>  
    print "dict['Alice']: ", dict['Alice'];  
KeyError: 'Alice'
```

## بروزرسانی دیکشنری ها ( Dictionaries )

در دیکشنری ها ( dictionaries ) شما می توانید داده ای را به دیکشنری اضافه نمایید و داده ها را بروزرسانی نمایید یا حتی عناصر را از دیکشنری حذف نمایید برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python  
  
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
dict['Age'] = 8; # update existing entry  
dict['School'] = "DPS School"; # Add new entry  
  
print "dict['Age']: ", dict['Age']  
print "dict['School']: ", dict['School']
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
dict['Age']: 8  
dict['School']: DPS School
```

## پاک کردن عناصر دیکشنری ها ( Dictionaries )

شما می توانید عناصر یک دیکشنری ( dictionary ) را هم به صورت تکی و هم به صورت گروهی حذف نمایید

برای پاک کردن تمام عناصر یک دیکشنری به صورت یکباره از تابع `del` استفاده می گردد برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python  
  
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
del dict['Name']; # remove entry with key 'Name'
```

```
dict.clear();      # remove all entries in dict
del dict ;        # delete entire dictionary

print "dict['Age']: ", dict['Age']
print "dict['School']: ", dict['School']
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
dict['Age']:
Traceback (most recent call last):
  File "test.py", line 8, in <module>
    print "dict['Age']: ", dict['Age'];
TypeError: 'type' object is unsubscriptable
```

## خصوصیات کلیدهای دیکشنری ها ( Dictionaries )

داده ها در دیکشنری ها ( dictionaries ) هیچ محدودیتی ندارند فقط باید دو نکته مهم را در مورد آنها مد نظر قرار دهید

نکته اول در دیکشنری ها ( dictionaries ) نمی توان دو کلید هم نام داشت و اگر چنین اتفاقی رخ بدهد داده برای آخرین کلید هم نام برای آن کلید در نظر گرفته می شود برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Name': 'Manni'}
print "dict['Name']: ", dict['Name']
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
dict['Name']: Manni
```

نکته دوم اینکه داده نمی تواند یک دیکشنری دیگر باشد برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7}
print "dict['Name']: ", dict['Name']
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Traceback (most recent call last):
  File "test.py", line 3, in <module>
    dict = {'Name': 'Zara', 'Age': 7};
```

```
TypeError: unhashable type: 'list'
```

## توابع و متدهای پیش ساخته برای دیکشنری ها ( Dictionaries )

در پایتون ( python ) یک سری توابع برای کار با دیکشنری ها وجود دارد که در زیر به همراه توضیحات آن آمده است

### توابع به همراه توضیحات

**cmp(dict1, dict2)**

از این متد برای مقایسه عناصر دو دیکشنری بکار می رود

```
#!/usr/bin/python
dict1 = {'Name': 'Zara', 'Age': 7};
dict2 = {'Name': 'Mahnaz', 'Age': 27};
dict3 = {'Name': 'Abid', 'Age': 27};
dict4 = {'Name': 'Zara', 'Age': 7};
print "Return Value : %d" % cmp (dict1, dict2)
print "Return Value : %d" % cmp (dict2, dict3)
print "Return Value : %d" % cmp (dict1, dict4)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Return Value : -1
Return Value : 1
Return Value : 0
```

**len(dict)**

این متد تعداد عناصر یک دیکشنری را بر می گرداند

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7};
print "Length : %d" % len (dict)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Length : 2
```



## **str(dict)**

این متد تمام عناصر یک دیکشنری را به صورت رشته بر می گرداند

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7};
print "Equivalent String : %s" % str (dict)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Equivalent String : {'Age': 7, 'Name': 'Zara'}
```

## **type(variable)**

این تابع نوع متغیر را بر می گرداند و اگر متغیر دیکشنری باشد کلمه dict را بر می گرداند

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7};
print "Variable Type : %s" % type (dict)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Variable Type : <type 'dict'>
```

پایتون (python) دارای یک سری متد برای کار با دیکشنری ها می باشد که در زیر به همراه توضیحات آمده اند

## **متدها به همراه توضیحات**

### **dict.clear()**

این متد همه عناصر یک دیکشنری را پاک می نماید

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7};
print "Start Len : %d" % len(dict)
dict.clear()
print "End Len : %d" % len(dict)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Start Len : 2
End Len : 0
```

### **dict.copy()**

این متد یک کپی از دیکشنری ایجاد می کند

```
#!/usr/bin/python

dict1 = {'Name': 'Zara', 'Age': 7};
dict2 = dict1.copy()
print "New Dictionary : %s" % str(dict2)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
New Dictionary : {'Age': 7, 'Name': 'Zara'}
```

### **dict.fromkeys()**

برای ایجاد یک دیکشنری جدید بکار می رود

```
#!/usr/bin/python

seq = ('name', 'age', 'sex')
dict = dict.fromkeys(seq)
print "New Dictionary : %s" % str(dict)

dict = dict.fromkeys(seq, 10)
print "New Dictionary : %s" % str(dict)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
New Dictionary : {'age': None, 'name': None, 'sex': None}
New Dictionary : {'age': 10, 'name': 10, 'sex': 10}
```

### **dict.get(key, default=None)**

این متد بر اساس کلیدی که دریافت می کند مقدار آن کلید را دریافت می نماید

```
#!/usr/bin/python
dict = {'Name': 'Zabra', 'Age': 7}
print "Value : %s" % dict.get('Age')
print "Value : %s" % dict.get('Education', "Never")
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : 7
Value : Never
```

### **dict.has\_key(key)**

این متد اگر کلید در دیکشنری وجود داشته باشد True را بر می گرداند در غیر این صورت False

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.has_key('Age')
print "Value : %s" % dict.has_key('Sex')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : True
Value : False
```

### **dict.items()**

تمام جفت داده و کلید های یک دیکشنری را به صورت لیست بر می گرداند

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.items()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : [('Age', 7), ('Name', 'Zara')]
```

### **dict.keys()**

تمام کلیدهای یک دیکشنری را به صورت لیست باز می گرداند

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.keys()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : ['Age', 'Name']
```

### **dict.setdefault(key, default=None)**

برای اضافه کردن یک کلید به دیکشنری بکار می رود

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.setdefault('Age', None)
print "Value : %s" % dict.setdefault('Sex', None)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : 7
Value : None
```

### **dict.update(dict2)**

از این متد برای اضافه کردن یک دیکشنری به انتهای دیکشنری دیگر بکار می رود

```
#!/usr/bin/python
```

```
dict = {'Name': 'Zara', 'Age': 7}
dict2 = {'Sex': 'female' }

dict.update(dict2)
print "Value : %s" % dict
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : {'Age': 7, 'Name': 'Zara', 'Sex': 'female'}
```

**dict.values()**

تمام داده های یک دیکشنری را به صورت لیست باز می گرداند

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.values()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value : [7, 'Zara']
```